**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра прикладной математики

ЛАБОРАТОРНАЯ РАБОТА №1

по современным платформам разработки программного обеспечения

«Разработка консольных java-приложений»

Студент \_\_\_\_\_\_\_\_\_ Шутов А.С.

Группа ПМ-21-2

Руководитель

к.т.н., доцент каф. ПМ \_\_\_\_\_\_\_\_\_ Мирошников А. И.

Липецк 2023 г.

**Цель работы**

Научиться создавать установочный файл для программы, написанной на языке C#.

**Задание**

Расширить лабораторную работу №4 «Механизм подготовки отчетов». Для существующей программы создать установочный файл.

**Краткое руководство пользователя**

Минимальные системные требования для запуска программы:

* Операционная система не ниже Windows 7;
* Не менее 1 ГБ оперативной памяти;
* Не менее 64 Мбайт видеопамяти;
* Не менее 16 ГБ свободного места на диске.

**Код программы**

Файл «MainWindow.axaml»

<Window xmlns=**"https://github.com/avaloniaui"**

xmlns:x=**"http://schemas.microsoft.com/winfx/2006/xaml"**

xmlns:vm=**"using:Avalonia.GUI.ViewModels"**

xmlns:d=**"http://schemas.microsoft.com/expression/blend/2008"**

xmlns:mc=**"http://schemas.openxmlformats.org/markup-compatibility/2006"**

xmlns:system=**"clr-namespace:System;assembly=System.Runtime"**

mc:Ignorable=**"d"** d:DesignWidth=**"600"** d:DesignHeight=**"500"**

MinWidth=**"{Binding WindowMinWidthProp}"**

MinHeight=**"{Binding WindowMinHeightProp}"**

Width=**"{Binding WindowWidthProp}"**

Height=**"{Binding WindowHeightProp}"**

CanResize=**"True"**

x:Class=**"Avalonia.GUI.Views.MainWindow"**

Icon=**"/Assets/sound-wave.ico"**

Title=**"Музыкальная библиотека"**

WindowStartupLocation=**"CenterScreen"**

Background=**"Transparent"**>

<Design.DataContext>

<vm:MainWindowViewModel/>

</Design.DataContext>

<Panel>

<ExperimentalAcrylicBorder IsHitTestVisible=**"False"**>

<ExperimentalAcrylicBorder.Material>

<ExperimentalAcrylicMaterial

BackgroundSource=**"Digger"**

TintColor=**"White"**

TintOpacity=**"1"**

MaterialOpacity=**"0.5"** />

</ExperimentalAcrylicBorder.Material>

</ExperimentalAcrylicBorder>

<Canvas Background=**"White"** Opacity=**"0.8"**/>

<!-- Главное окно входа -->

<!-- IsVisible="{Binding MainLoginWindowState}" -->

<Grid

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**

IsVisible=**"{Binding MainLoginWindowState}"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"70"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"150"**></ColumnDefinition>

<ColumnDefinition Width=**"150"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image

Grid.Column=**"0"**

Grid.Row=**"0"**

Grid.ColumnSpan=**"2"**

MaxWidth=**"50"**

VerticalAlignment=**"Top"**

Source=**"/Assets/sound-wave.ico"**/>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"1"**

Grid.ColumnSpan=**"2"**

FontSize=**"20"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Text=**"Вход"**

FontWeight=**"DemiBold"**/>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"2"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Text=**"Перейти в Музыкальную библиотеку"**/>

<TextBox

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"4"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

Watermark=**"Имя пользователя"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"20"**

Text=**"{Binding UsernameProp}"**>

</TextBox>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"5"**

Grid.ColumnSpan=**"2"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"{Binding UserExistsWarningProp}"**/>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Padding=**"5, 5, 5, 5"**

Grid.Column=**"0"**

Grid.Row=**"6"**

FontSize=**"15"**

FontWeight=**"Bold"**

Background=**"Transparent"**

Command=**"{Binding LoginWindowChange}"**>

<Button.CommandParameter>

<system:String>**Создать аккаунт**</system:String>

</Button.CommandParameter>

**Создать аккаунт**

</Button>

<Button

CornerRadius=**"3"**

Padding=**"15, 5, 15, 5"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Grid.Column=**"1"**

Grid.Row=**"6"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding UserExistsCheck}"**>

**Далее**

</Button>

</Grid>

<!-- Ввод пароля -->

<!-- IsVisible="{Binding PasswordEntryWindowState}" -->

<Grid

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**

IsVisible=**"{Binding PasswordEntryWindowState}"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"70"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"150"**></ColumnDefinition>

<ColumnDefinition Width=**"150"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image

Grid.Column=**"0"**

Grid.Row=**"0"**

Grid.ColumnSpan=**"2"**

MaxWidth=**"50"**

VerticalAlignment=**"Top"**

Source=**"/Assets/sound-wave.ico"**/>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"1"**

Grid.ColumnSpan=**"2"**

FontSize=**"20"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Text=**"Добро пожаловать!"**

FontWeight=**"DemiBold"**/>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"2"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Text=**"{Binding UsernameProp}"**/>

<TextBox

Classes=**"revealPasswordButton"**

PasswordChar=**"•"**

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"4"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

Watermark=**"Введите пароль"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"20"**

Text=**"{Binding PasswordEntryProp}"**>

</TextBox>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"5"**

Grid.ColumnSpan=**"2"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"{Binding PasswordEntryWarningProp}"**/>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Padding=**"5, 5, 5, 5"**

Grid.Column=**"0"**

Grid.Row=**"6"**

FontSize=**"15"**

FontWeight=**"Bold"**

Background=**"Transparent"**

Command=**"{Binding LoginWindowChange}"**>

<Button.CommandParameter>

<system:String>**Войти**</system:String>

</Button.CommandParameter>

**← Назад**

</Button>

<Button

CornerRadius=**"3"**

Padding=**"15, 5, 15, 5"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Grid.Column=**"1"**

Grid.Row=**"6"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding PasswordEntryCheck}"**>

**Далее**

</Button>

</Grid>

<!-- Создание аккаунта -->

<!-- IsVisible="{Binding CreateAccountWindowState}" -->

<Grid

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**

IsVisible=**"{Binding CreateAccountWindowState}"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"50"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"50"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"150"**></ColumnDefinition>

<ColumnDefinition Width=**"150"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image

Grid.Column=**"0"**

Grid.Row=**"0"**

Grid.ColumnSpan=**"2"**

MaxWidth=**"50"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Source=**"/Assets/sound-wave.ico"**/>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"1"**

Grid.ColumnSpan=**"2"**

FontSize=**"20"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Text=**"Создайте аккаунт"**

FontWeight=**"DemiBold"**/>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"2"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"Перейти в Музыкальную библиотеку"**/>

<TextBox

Text = **"{Binding NameProp}"**

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"3"**

FontSize=**"15"**

Watermark=**"Имя"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Width=**"145"**

Height=**"20"**/>

<TextBox

Text = **"{Binding SurnameProp}"**

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"1"**

Grid.Row=**"3"**

FontSize=**"15"**

Watermark=**"Фамилия"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Width=**"145"**

Height=**"20"**/>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"4"**

Grid.ColumnSpan=**"2"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"{Binding NameSurnameWarningProp}"**/>

<TextBox

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"5"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

Watermark=**"Имя пользователя"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"20"**

Text=**"{Binding UsernameProp}"**/>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"6"**

Grid.ColumnSpan=**"2"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"{Binding UsernameWarningAlignmentProp}"**

Text=**"{Binding UsernameWarningProp}"**/>

<TextBox

Text=**"{Binding PasswordProp}"**

Classes=**"revealPasswordButton"**

PasswordChar=**"•"**

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"7"**

FontSize=**"15"**

Watermark=**"Пароль"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Width=**"145"**

Height=**"20"**/>

<TextBox

Text=**"{Binding PasswordConfirmationProp}"**

Classes=**"revealPasswordButton"**

PasswordChar=**"•"**

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"1"**

Grid.Row=**"7"**

FontSize=**"15"**

Watermark=**"Подтвердить"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Width=**"145"**

Height=**"20"**/>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"{Binding PasswordWarningAlignmentProp}"**

Text=**"{Binding PasswordWarningProp}"**/>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Padding=**"5, 5, 5, 5"**

Grid.Column=**"0"**

Grid.Row=**"9"**

FontSize=**"15"**

FontWeight=**"Bold"**

Background=**"Transparent"**

Command=**"{Binding LoginWindowChange}"**>

<Button.CommandParameter>

<system:String>**Войти**</system:String>

</Button.CommandParameter>

**Войти**

</Button>

<Button

CornerRadius=**"3"**

Padding=**"15, 5, 15, 5"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Grid.Column=**"1"**

Grid.Row=**"9"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding NextCreateAccount}"**>

**Далее**

</Button>

</Grid>

<!-- Аккаунт создан -->

<!-- IsVisible="{Binding AccountCreatedWindowState}" -->

<Grid

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**

IsVisible=**"{Binding AccountCreatedWindowState}"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"70"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"70"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"200"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image

Grid.Column=**"0"**

Grid.Row=**"0"**

MaxWidth=**"50"**

VerticalAlignment=**"Top"**

Source=**"/Assets/sound-wave.ico"**/>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"1"**

FontSize=**"15"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Text=**"✓ Ваш аккаунт создан"**/>

<Button

CornerRadius=**"3"**

Padding=**"15, 5, 15, 5"**

VerticalAlignment=**"Bottom"**

HorizontalAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"2"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding NextAccountCreated}"**>

**Далее**

</Button>

</Grid>

<!-- Главное окно -->

<!-- IsVisible="{Binding MainWindowState}" -->

<Grid

IsVisible=**"{Binding MainWindowState}"**

HorizontalAlignment=**"Center"**

Width=**"900"**

Height=**"600"**

VerticalAlignment=**"Center"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"50"**></RowDefinition>

<RowDefinition Height=**"500"**></RowDefinition>

<RowDefinition Height=**"50"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"275"**></ColumnDefinition>

<ColumnDefinition Width=**"350"**></ColumnDefinition>

<ColumnDefinition Width=**"275"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Grid.Row=**"0"**

Grid.Column=**"0"**

Padding=**"5, 5, 5, 5"**

Background=**"Transparent"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding LoginWindowChange}"**>

<Button.CommandParameter>

<system:String>**Выйти**</system:String>

</Button.CommandParameter>

**← Выйти**

</Button>

<StackPanel

Orientation=**"Horizontal"**

Grid.Row=**"0"**

Grid.Column=**"1"**

Grid.ColumnSpan=**"2"**

Spacing=**"3"**

HorizontalAlignment=**"Right"**

VerticalAlignment=**"Center"**>

<Image

Height=**"20"**

Source=**"/Assets/user-icon.png"**/>

<TextBlock

FontSize=**"15"**

Text=**"{Binding UsernameProp}"**/>

</StackPanel>

<DataGrid

Grid.Row=**"1"**

Grid.Column=**"0"**

Grid.ColumnSpan=**"3"**

FontSize=**"15"**

Background=**"#F5F5F5"**

VerticalScrollBarVisibility=**"Visible"**

Height=**"500"**

SelectedItem=**"{Binding SelectedTrack}"**

Items=**"{Binding Tracks}"**

AutoGenerateColumns=**"False"**

SelectionMode=**"Single"**

RowHeight=**"100"**

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**>

<DataGrid.Styles>

<Style Selector=**"DataGridCell:current /template/ Grid#FocusVisual"**>

<Setter Property=**"IsVisible"** Value=**"False"** />

</Style>

<Style Selector=**"DataGridColumnHeader"**>

<Setter Property=**"FontSize"** Value=**"15"**/>

<Setter Property=**"Foreground"** Value=**"Black"**/>

<Setter Property=**"Background"** Value=**"#EBEBEB"**/>

<Setter Property=**"HorizontalContentAlignment"** Value=**"Left"**/>

</Style>

</DataGrid.Styles>

<DataGrid.Columns>

<DataGridTemplateColumn

CanUserSort=**"False"**

Width=**"50"**>

<DataGridTemplateColumn.HeaderTemplate>

<DataTemplate>

<Image

Margin=**"12"**

HorizontalAlignment=**"Center"**

Width=**"25"**

Source=**"/Assets/heart-fill.png"**></Image>

</DataTemplate>

</DataGridTemplateColumn.HeaderTemplate>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Panel>

<Panel.Styles>

<Style Selector=**"ToggleButton:indeterminate"**>

<Setter Property=**"Opacity"** Value=**"0"**/>

<Setter Property=**"Foreground"** Value=**"Transparent"**/>

<Setter Property=**"Background"** Value=**"Transparent"**/>

</Style>

<Style Selector=**"ToggleButton:checked"**>

<Setter Property=**"Opacity"** Value=**"0"**/>

<Setter Property=**"Foreground"** Value=**"Transparent"**/>

<Setter Property=**"Background"** Value=**"Transparent"**/>

</Style>

<Style Selector=**"ToggleButton:unchecked"**>

<Setter Property=**"Foreground"** Value=**"Transparent"**/>

<Setter Property=**"Background"** Value=**"Transparent"**/>

</Style>

</Panel.Styles>

<Image

IsVisible=**"{Binding !Darling}"**

Width=**"30"**

HorizontalAlignment=**"Center"**

Source=**"/Assets/heart-empty.png"**/>

<Image

IsVisible=**"{Binding Darling}"**

Width=**"30"**

HorizontalAlignment=**"Center"**

Source=**"/Assets/heart-fill.png"**/>

<ToggleButton

Width=**"30"**

Height=**"30"**

HorizontalAlignment=**"Center"**

IsChecked=**"{Binding Darling, Mode=TwoWay}"**>

</ToggleButton>

</Panel>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn

IsReadOnly=**"True"**

CanUserSort=**"False"**

Header=**"Обложка"**

Width=**"115"**>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Source=**"{Binding Cover}"**></Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn

IsReadOnly=**"True"**

CanUserSort=**"False"**

Header=**"Название"**

FontSize=**"15"**

FontWeight=**"Bold"**

Width=**"175"**

Binding=**"{Binding Title}"**/>

<DataGridTextColumn

IsReadOnly=**"True"**

CanUserSort=**"False"**

Header=**"Артист"**

FontSize=**"15"**

FontWeight=**"Bold"**

Width=**"175"**

Binding=**"{Binding Artist}"**/>

<DataGridTextColumn

IsReadOnly=**"True"**

CanUserSort=**"False"**

Header=**"Дата релиза"**

FontSize=**"15"**

FontWeight=**"Bold"**

Width=**"175"**

Binding=**"{Binding ReleaseDate}"**/>

<DataGridTextColumn

IsReadOnly=**"True"**

CanUserSort=**"False"**

Header=**"Последнее изменение"**

FontSize=**"15"**

FontWeight=**"Bold"**

Width=**"210"**

Binding=**"{Binding LastChangeTime}"**/>

</DataGrid.Columns>

</DataGrid>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Grid.Column=**"1"**

Grid.Row=**"2"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding AddTrack}"**>

<Button.CommandParameter>

<system:String>**Добавить**</system:String>

</Button.CommandParameter>

**Добавить**

</Button>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Grid.Column=**"1"**

Grid.Row=**"2"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding EditTrack}"**>

<Button.CommandParameter>

<system:String>**Изменить**</system:String>

</Button.CommandParameter>

**Изменить**

</Button>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Grid.Column=**"1"**

Grid.Row=**"2"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding DelTrack}"**>

**Удалить**

</Button>

</Grid>

<!-- Изменить трек -->

<!-- IsVisible="{Binding EditTrackWindow}" -->

<Panel

IsVisible=**"{Binding EditTrackWindow}"**>

<Canvas Background=**"Black"** Opacity=**"0.25"**/>

<Grid

Background=**"#EBEBEB"**

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"100"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"250"**></ColumnDefinition>

<ColumnDefinition Width=**"250"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"0"**

Grid.ColumnSpan=**"2"**

FontSize=**"20"**

Margin=**"100, 10, 0, 0"**

VerticalAlignment=**"Bottom"**

HorizontalAlignment=**"Left"**

Text=**"Изменить трек"**

FontWeight=**"DemiBold"**/>

<TextBox

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"2"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

Watermark=**"Название"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"20"**

Text=**"{Binding SelectedTrack.Title}"**>

</TextBox>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"3"**

Grid.ColumnSpan=**"2"**

Margin=**"100, 0, 0, 0"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"{Binding TitleWarning}"**/>

<TextBox

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"4"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

Watermark=**"Артист"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"20"**

Text=**"{Binding SelectedTrack.Artist}"**>

</TextBox>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"5"**

Grid.ColumnSpan=**"2"**

Margin=**"100, 0, 0, 0"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"{Binding ArtistWarning}"**/>

<DatePicker

FontWeight=**"Medium"**

CornerRadius=**"3"**

BorderThickness=**"1"**

Grid.Column=**"0"**

Grid.Row=**"6"**

Grid.ColumnSpan=**"2"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"35"**

SelectedDate=**"{Binding TmpReleaseDate}"**/>

<TextBlock

Background=**"White"**

TextAlignment=**"Center"**

Margin=**"100, 0, 0, 0"**

Padding=**"0, 23, 0, 0"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Height=**"100"**

Width=**"100"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

TextWrapping=**"Wrap"**

Text=**"Здесь будет выбранная обложка"**/>

<Image

Margin=**"100, 0, 0, 0"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Height=**"100"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

Source=**"{Binding TmpCover}"**/>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

Margin=**"0, 50, 147, 0"**

FontSize=**"10"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Text=**"{Binding CoverWarning}"**/>

<Button

Margin=**"0, 0, 100, 0"**

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Center"**

VerticalContentAlignment=**"Center"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

FontWeight=**"Bold"**

Content=**"Выбрать обложку"**

Click=**"Button\_OnClick"**>

</Button>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Grid.Column=**"0"**

Grid.Row=**"10"**

Margin=**"100, 0, 0, 0"**

FontSize=**"15"**

FontWeight=**"Bold"**

Background=**"Transparent"**

Command=**"{Binding EditTrack}"**>

<Button.CommandParameter>

<system:String>**Назад**</system:String>

</Button.CommandParameter>

**← Назад**

</Button>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Right"**

Grid.Column=**"1"**

Grid.Row=**"10"**

Margin=**"0, 0, 100, 0"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding EditTrack}"**>

<Button.CommandParameter>

<system:String>**Готово**</system:String>

</Button.CommandParameter>

**Готово**

</Button>

</Grid>

</Panel>

<!-- Добавить трек -->

<!-- IsVisible="{Binding AddTrackWindow}" -->

<Panel

IsVisible=**"{Binding AddTrackWindow}"**>

<Canvas Background=**"Black"** Opacity=**"0.25"**/>

<Grid

Background=**"#EBEBEB"**

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"100"**></RowDefinition>

<RowDefinition Height=**"30"**></RowDefinition>

<RowDefinition Height=**"40"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"250"**></ColumnDefinition>

<ColumnDefinition Width=**"250"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock

Grid.Column=**"0"**

Grid.Row=**"0"**

Grid.ColumnSpan=**"2"**

FontSize=**"20"**

Margin=**"100, 10, 0, 0"**

VerticalAlignment=**"Bottom"**

HorizontalAlignment=**"Left"**

Text=**"Добавить трек"**

FontWeight=**"DemiBold"**/>

<TextBox

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"2"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

Watermark=**"Название"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"20"**

Text=**"{Binding TmpTitle}"**>

</TextBox>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"3"**

Grid.ColumnSpan=**"2"**

Margin=**"100, 0, 0, 0"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"{Binding TitleWarning}"**/>

<TextBox

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Left"**

VerticalContentAlignment=**"Center"**

Grid.Column=**"0"**

Grid.Row=**"4"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

Watermark=**"Артист"**

FontWeight=**"Medium"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"20"**

Text=**"{Binding TmpArtist}"**>

</TextBox>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"5"**

Grid.ColumnSpan=**"2"**

Margin=**"100, 0, 0, 0"**

FontSize=**"10"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Text=**"{Binding ArtistWarning}"**/>

<DatePicker

FontWeight=**"Medium"**

CornerRadius=**"3"**

BorderThickness=**"1"**

Grid.Column=**"0"**

Grid.Row=**"6"**

Grid.ColumnSpan=**"2"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

Width=**"300"**

Height=**"35"**

SelectedDate=**"{Binding TmpReleaseDate}"**/>

<TextBlock

Background=**"White"**

TextAlignment=**"Center"**

Margin=**"100, 0, 0, 0"**

Padding=**"0, 23, 0, 0"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Height=**"100"**

Width=**"100"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

TextWrapping=**"Wrap"**

Text=**"Здесь будет выбранная обложка"**/>

<Image

Margin=**"100, 0, 0, 0"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Height=**"100"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

Source=**"{Binding TmpCover}"**/>

<TextBlock

TextWrapping=**"Wrap"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

Margin=**"0, 50, 147, 0"**

FontSize=**"10"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Text=**"{Binding CoverWarning}"**/>

<Button

Margin=**"0, 0, 100, 0"**

CornerRadius=**"3"**

BorderThickness=**"1"**

HorizontalContentAlignment=**"Center"**

VerticalContentAlignment=**"Center"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Right"**

Grid.Column=**"0"**

Grid.Row=**"8"**

Grid.ColumnSpan=**"2"**

FontSize=**"15"**

FontWeight=**"Bold"**

Content=**"Выбрать обложку"**

Click=**"Button\_OnClick"**>

</Button>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Left"**

Grid.Column=**"0"**

Grid.Row=**"10"**

Margin=**"100, 0, 0, 0"**

FontSize=**"15"**

FontWeight=**"Bold"**

Background=**"Transparent"**

Command=**"{Binding AddTrack}"**>

<Button.CommandParameter>

<system:String>**Назад**</system:String>

</Button.CommandParameter>

**← Назад**

</Button>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Top"**

HorizontalAlignment=**"Right"**

Grid.Column=**"1"**

Grid.Row=**"10"**

Margin=**"0, 0, 100, 0"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding AddTrack}"**>

<Button.CommandParameter>

<system:String>**Готово**</system:String>

</Button.CommandParameter>

**Готово**

</Button>

</Grid>

</Panel>

<!-- Админка -->

<!-- IsVisible="{Binding AdminWindowState}" -->

<Grid

IsVisible=**"{Binding AdminWindowState}"**

HorizontalAlignment=**"Center"**

Width=**"500"**

Height=**"460"**

VerticalAlignment=**"Center"**>

<Grid.RowDefinitions>

<RowDefinition Height=**"50"**></RowDefinition>

<RowDefinition Height=**"300"**></RowDefinition>

<RowDefinition Height=**"10"**></RowDefinition>

<RowDefinition Height=**"50"**></RowDefinition>

<RowDefinition Height=**"50"**></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width=**"50"**></ColumnDefinition>

<ColumnDefinition Width=**"400"**></ColumnDefinition>

<ColumnDefinition Width=**"50"**></ColumnDefinition>

</Grid.ColumnDefinitions>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Left"**

Grid.Row=**"0"**

Grid.Column=**"1"**

Padding=**"5, 5, 5, 5"**

Background=**"Transparent"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding LoginWindowChange}"**>

<Button.CommandParameter>

<system:String>**Выйти**</system:String>

</Button.CommandParameter>

**← Выйти**

</Button>

<StackPanel

Orientation=**"Horizontal"**

Grid.Row=**"0"**

Grid.Column=**"1"**

Spacing=**"3"**

HorizontalAlignment=**"Right"**

VerticalAlignment=**"Center"**>

<Image

Height=**"20"**

Source=**"/Assets/user-icon.png"**/>

<TextBlock

FontSize=**"15"**

Text=**"administrator"**/>

</StackPanel>

<DataGrid

Grid.Row=**"1"**

Grid.Column=**"0"**

Grid.ColumnSpan=**"3"**

FontSize=**"15"**

Background=**"#F5F5F5"**

VerticalScrollBarVisibility=**"Visible"**

Height=**"300"**

Items=**"{Binding Users}"**

AutoGenerateColumns=**"False"**

SelectionMode=**"Single"**

RowHeight=**"30"**

HorizontalAlignment=**"Center"**

VerticalAlignment=**"Center"**>

<DataGrid.Styles>

<Style Selector=**"DataGridCell:current /template/ Grid#FocusVisual"**>

<Setter Property=**"IsVisible"** Value=**"False"** />

</Style>

<Style Selector=**"DataGridColumnHeader"**>

<Setter Property=**"FontSize"** Value=**"15"**/>

<Setter Property=**"Foreground"** Value=**"Black"**/>

<Setter Property=**"Background"** Value=**"#EBEBEB"**/>

<Setter Property=**"HorizontalContentAlignment"** Value=**"Center"**/>

</Style>

</DataGrid.Styles>

<DataGrid.Columns>

<DataGridTemplateColumn

CanUserSort=**"False"**

Width=**"100"**>

<DataGridTemplateColumn.HeaderTemplate>

<DataTemplate>

<Image

Margin=**"13, 0, 0, 0"**

HorizontalAlignment=**"Center"**

Width=**"20"**

Source=**"/Assets/checkbox-symbol.png"**></Image>

</DataTemplate>

</DataGridTemplateColumn.HeaderTemplate>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<CheckBox

Width=**"30"**

Height=**"30"**

HorizontalAlignment=**"Center"**

IsChecked=**"{Binding Selected}"**>

</CheckBox>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn

IsReadOnly=**"True"**

CanUserSort=**"False"**

Header=**"Пользователь"**

FontSize=**"15"**

FontWeight=**"Bold"**

Width=**"300"**

Binding=**"{Binding Username}"**/>

</DataGrid.Columns>

</DataGrid>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

HorizontalContentAlignment=**"Center"**

Grid.Column=**"1"**

Grid.Row=**"3"**

Width=**"280"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding CreateCustomReport}"**>

**Создать пользовательский отчет**

</Button>

<Button

CornerRadius=**"3"**

VerticalAlignment=**"Center"**

HorizontalAlignment=**"Center"**

HorizontalContentAlignment=**"Center"**

Grid.Column=**"1"**

Grid.Row=**"4"**

Width=**"280"**

FontSize=**"15"**

FontWeight=**"Bold"**

Command=**"{Binding CreateIntegratedReport}"**>

**Создать интегральный отчет**

</Button>

</Grid>

</Panel>

</Window>

Файл «MainWindowViewModel.cs»

**using** ReactiveUI**;**

**using** System**;**

**using** System**.**IO**;**

**using** System**.**Drawing**;**

**using** System**.**Diagnostics**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**Collections**.**ObjectModel**;**

**using** System**.**Data**;**

**using** System**.**Text**;**

**using** Microsoft**.**CodeAnalysis**.**CSharp**.**Syntax**;**

**using** System**.**Data**.**SQLite**;**

**using** System**.**Linq**;**

**using** System**.**Text**.**RegularExpressions**;**

**using** Avalonia**.**Controls**;**

**using** Avalonia**.**Interactivity**;**

**using** Avalonia**.**Media**.**Imaging**;**

**using** OfficeOpenXml**;**

**using** OfficeOpenXml**.**Style**;**

**using** ReactiveUI**.**Fody**.**Helpers**;**

**namespace** Avalonia**.**GUI**.**ViewModels

**{**

**public** class MainWindowViewModel **:** ViewModelBase

**{**

**public** void SetImagePath**(**string path**)**

**{**

TmpCover **=** **new** Avalonia**.**Media**.**Imaging**.**Bitmap**(new** FileStream**(**path**,** FileMode**.**Open**));**

**}**

**[**Reactive**]** **public** Track SelectedTrack **{** **get;** **set;** **}**

**[**Reactive**]** **private** bool EditTrackWindow **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** bool AddTrackWindow **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** bool TmpWarning **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** string TitleWarning **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string ArtistWarning **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string CoverWarning **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string TmpTitle **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string TmpArtist **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** bool TmpDarling **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** Avalonia**.**Media**.**Imaging**.**Bitmap TmpCover **{** **get;** **set;** **}** **=** **null;**

**[**Reactive**]** **private** DateTimeOffset TmpReleaseDate **{** **get;** **set;** **}** **=** DateTimeOffset**.**Now**;**

**[**Reactive**]** **private** string TmpLastChangeTime **{** **get;** **set;** **}** **=** ""**;**

**public** class Track **:** ViewModelBase

**{**

**public** string Title **{** **get;** **set;** **}**

**public** string Artist **{** **get;** **set;** **}**

**[**Reactive**]** **public** bool Darling **{** **get;** **set;** **}**

**public** Avalonia**.**Media**.**Imaging**.**Bitmap Cover **{** **get;** **set;** **}**

**public** string ReleaseDate **{** **get;** **set;** **}**

**public** string LastChangeTime **{** **get;** **set;** **}**

**}**

**private** ObservableCollection**<**Track**>** Tracks **{** **get;** **}**

**public** class TrackForReport **:** ViewModelBase

**{**

**public** string Title **{** **get;** **set;** **}**

**public** string Artist **{** **get;** **set;** **}**

**public** string ReleaseDate **{** **get;** **set;** **}**

**public** int Count **{** **get;** **set;** **}**

**}**

**private** ObservableCollection**<**TrackForReport**>** TracksForReport **{** **get;** **}**

**private** ObservableCollection**<**TrackForReport**>** SortedTracksForReport **{** **get;** **}**

**public** MainWindowViewModel**()**

**{**

Tracks **=** **new** ObservableCollection**<**Track**>();**

TracksForReport **=** **new** ObservableCollection**<**TrackForReport**>();**

SortedTracksForReport **=** **new** ObservableCollection**<**TrackForReport**>();**

Users **=** **new** ObservableCollection**<**User**>();**

**}**

**private** void EditTrack**(**string command**)**

**{**

**if** **(**command **==** "Назад"**)**

**{**

EditTrackWindow **=** **false;**

TmpWarning **=** **false;**

TmpTitle **=** ""**;**

TmpArtist **=** ""**;**

TmpDarling **=** **false;**

TmpCover **=** **null;**

TmpReleaseDate **=** DateTimeOffset**.**Now**;**

TmpLastChangeTime **=** ""**;**

TitleWarning **=** ""**;**

ArtistWarning **=** ""**;**

CoverWarning **=** ""**;**

**}**

**if** **(**command **==** "Изменить" **&&** SelectedTrack **!=** **null)**

**{**

EditTrackWindow **=** **true;**

TmpReleaseDate **=** DateTimeOffset**.**Parse**(**SelectedTrack**.**ReleaseDate**);**

TmpCover **=** SelectedTrack**.**Cover**;**

**}**

**if** **(**command **==** "Готово"**)**

**{**

TmpWarning **=** **false;**

TitleWarning **=** ""**;**

ArtistWarning **=** ""**;**

**if** **(**string**.**IsNullOrEmpty**(**SelectedTrack**.**Title**))**

**{**

TitleWarning **=** "⚠ Введите название трека"**;**

TmpWarning **=** **true;**

**}**

**if** **(**string**.**IsNullOrEmpty**(**SelectedTrack**.**Artist**))**

**{**

ArtistWarning **=** "⚠ Введите имя артиста"**;**

TmpWarning **=** **true;**

**}**

**if** **(**TmpWarning**)**

**{**

**return;**

**}**

SelectedTrack**.**Cover **=** TmpCover**;**

SelectedTrack**.**ReleaseDate **=** TmpReleaseDate**.**ToString**(**"dd MMMM, yyyy"**);**

SelectedTrack**.**LastChangeTime **=** DateTime**.**Now**.**ToString**(**"HH:mm:ss dd.MM.yy"**);**

ObservableCollection**<**Track**>** Tmp **=** **new** ObservableCollection**<**Track**>(**Tracks**);**

Tracks**.**Clear**();**

**for** **(**int i **=** 0**;** i **<** Tmp**.**Count**;** i**++)**

**{**

Tracks**.**Add**(**

**new** Track**()**

**{**

Title **=** Tmp**[**i**].**Title**,**

Artist **=** Tmp**[**i**].**Artist**,**

Darling **=** Tmp**[**i**].**Darling**,**

Cover **=** Tmp**[**i**].**Cover**,**

ReleaseDate **=** Tmp**[**i**].**ReleaseDate**,**

LastChangeTime **=** Tmp**[**i**].**LastChangeTime

**}**

**);**

**}**

Tmp**.**Clear**();**

EditTrackWindow **=** **false;**

TitleWarning **=** ""**;**

ArtistWarning **=** ""**;**

CoverWarning **=** ""**;**

**}**

**}**

**private** void AddTrack**(**string command**)**

**{**

**if** **(**command **==** "Назад"**)**

**{**

AddTrackWindow **=** **false;**

TmpWarning **=** **false;**

TmpTitle **=** ""**;**

TmpArtist **=** ""**;**

TmpDarling **=** **false;**

TmpCover **=** **null;**

TmpReleaseDate **=** DateTimeOffset**.**Now**;**

TmpLastChangeTime **=** ""**;**

TitleWarning **=** ""**;**

ArtistWarning **=** ""**;**

CoverWarning **=** ""**;**

**}**

**if** **(**command **==** "Добавить"**)**

**{**

AddTrackWindow **=** **true;**

**}**

**if** **(**command **==** "Готово"**)**

**{**

TmpWarning **=** **false;**

TitleWarning **=** ""**;**

ArtistWarning **=** ""**;**

CoverWarning **=** ""**;**

**if** **(**string**.**IsNullOrEmpty**(**TmpTitle**))**

**{**

TitleWarning **=** "⚠ Введите название трека"**;**

TmpWarning **=** **true;**

**}**

**if** **(**string**.**IsNullOrEmpty**(**TmpArtist**))**

**{**

ArtistWarning **=** "⚠ Введите имя артиста"**;**

TmpWarning **=** **true;**

**}**

**if** **(**TmpCover **==** **null)**

**{**

CoverWarning **=** "⚠ Выберите обложку"**;**

TmpWarning **=** **true;**

**}**

**if** **(**TmpWarning**)**

**{**

**return;**

**}**

Tracks**.**Add**(**

**new** Track**()**

**{**

Title **=** TmpTitle**,**

Artist **=** TmpArtist**,**

Darling **=** TmpDarling**,**

Cover **=** TmpCover**,**

ReleaseDate **=** TmpReleaseDate**.**ToString**(**"dd MMMM, yyyy"**),**

LastChangeTime **=** DateTime**.**Now**.**ToString**(**"HH:mm:ss dd.MM.yy"**)**

**}**

**);**

AddTrackWindow **=** **false;**

TmpTitle **=** ""**;**

TmpArtist **=** ""**;**

TmpCover **=** **null;**

TmpDarling **=** **false;**

TitleWarning **=** ""**;**

ArtistWarning **=** ""**;**

CoverWarning **=** ""**;**

**}**

**}**

**private** void DelTrack**()**

**{**

**if** **(**SelectedTrack **!=** **null)**

**{**

Tracks**.**Remove**(**SelectedTrack**);**

**}**

**}**

**[**Reactive**]** **private** int WindowWidthProp **{** **get;** **set;** **}** **=** 600**;**

**[**Reactive**]** **private** int WindowHeightProp **{** **get;** **set;** **}** **=** 500**;**

**[**Reactive**]** **private** int WindowMinWidthProp **{** **get;** **set;** **}** **=** 600**;**

**[**Reactive**]** **private** int WindowMinHeightProp **{** **get;** **set;** **}** **=** 500**;**

**[**Reactive**]** **private** bool MainLoginWindowState **{** **get;** **set;** **}** **=** **true;**

**[**Reactive**]** **private** bool CreateAccountWindowState **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** bool AccountCreatedWindowState **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** bool PasswordEntryWindowState **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** bool MainWindowState **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** bool AdminWindowState **{** **get;** **set;** **}** **=** **false;**

**[**Reactive**]** **private** string PasswordEntryProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string NameProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string SurnameProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string NameSurnameWarningProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string UsernameWarningProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string NameSurnameWarningAlignmentProp **{** **get;** **set;** **}** **=** "Center"**;**

**[**Reactive**]** **private** string UsernameWarningAlignmentProp **{** **get;** **set;** **}** **=** "Center"**;**

**[**Reactive**]** **private** string PasswordWarningProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string PasswordWarningAlignmentProp **{** **get;** **set;** **}** **=** "Center"**;**

**[**Reactive**]** **private** string UsernameProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string PasswordProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string PasswordConfirmationProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string UserExistsWarningProp **{** **get;** **set;** **}** **=** ""**;**

**[**Reactive**]** **private** string PasswordEntryWarningProp **{** **get;** **set;** **}** **=** ""**;**

**public** class User **:** ViewModelBase

**{**

**public** string Username **{** **get;** **set;** **}**

**[**Reactive**]** **public** bool Selected **{** **get;** **set;** **}**

**}**

**private** ObservableCollection**<**User**>** Users **{** **get;** **}**

**private** void LoginWindowChange**(**string SelectedWindow**)**

**{**

**if** **(!**File**.**Exists**(**@"db.sqlite3"**))**

**{**

SQLiteConnection**.**CreateFile**(**@"db.sqlite3"**);**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** "CREATE TABLE IF NOT EXISTS [accounts] ([id] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [name] TEXT NOT NULL, [surname] TEXT NOT NULL, " **+**

"[username] TEXT NOT NULL, [password] TEXT NOT NULL)"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

Command**.**ExecuteNonQuery**();**

commandText **=** "CREATE TABLE IF NOT EXISTS [data] ([id] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [owner] TEXT NOT NULL, [title] TEXT NOT NULL, " **+**

"[artist] TEXT NOT NULL)"**;**

Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Command**.**ExecuteNonQuery**();**

Connect**.**Close**();**

**}**

**}**

**if** **(**SelectedWindow **==** "Создать аккаунт"**)**

**{**

MainLoginWindowState **=** **false;**

CreateAccountWindowState **=** **true;**

AccountCreatedWindowState **=** **false;**

PasswordEntryWindowState **=** **false;**

MainWindowState **=** **false;**

AdminWindowState **=** **false;**

UserExistsWarningProp **=** ""**;**

NameProp **=** ""**;**

SurnameProp **=** ""**;**

UsernameProp **=** ""**;**

PasswordProp **=** ""**;**

PasswordConfirmationProp **=** ""**;**

NameSurnameWarningProp **=** ""**;**

NameSurnameWarningAlignmentProp **=** "Center"**;**

UsernameWarningProp **=** "Можно использовать буквы латинского алфавита, цифры и точки."**;**

UsernameWarningAlignmentProp **=** "Center"**;**

PasswordWarningProp **=** "Пароль должен содержать не менее восьми знаков, включать буквы, цифры и специальные символы"**;**

PasswordWarningAlignmentProp **=** "Center"**;**

**}**

**else** **if** **(**SelectedWindow **==** "Войти"**)**

**{**

UserExistsWarningProp **=** ""**;**

NameProp **=** ""**;**

SurnameProp **=** ""**;**

UsernameProp **=** ""**;**

PasswordProp **=** ""**;**

PasswordConfirmationProp **=** ""**;**

PasswordEntryProp **=** ""**;**

MainLoginWindowState **=** **true;**

CreateAccountWindowState **=** **false;**

AccountCreatedWindowState **=** **false;**

PasswordEntryWindowState **=** **false;**

MainWindowState **=** **false;**

AdminWindowState **=** **false;**

WindowWidthProp **=** 600**;**

WindowMinWidthProp **=** 600**;**

WindowHeightProp **=** 500**;**

WindowMinHeightProp **=** 500**;**

Tracks**.**Clear**();**

Users**.**Clear**();**

**}**

**else** **if** **(**SelectedWindow **==** "Выйти"**)**

**{**

DatabaseUpdate**();**

UserExistsWarningProp **=** ""**;**

NameProp **=** ""**;**

SurnameProp **=** ""**;**

UsernameProp **=** ""**;**

PasswordProp **=** ""**;**

PasswordConfirmationProp **=** ""**;**

PasswordEntryProp **=** ""**;**

MainLoginWindowState **=** **true;**

CreateAccountWindowState **=** **false;**

AccountCreatedWindowState **=** **false;**

PasswordEntryWindowState **=** **false;**

MainWindowState **=** **false;**

AdminWindowState **=** **false;**

WindowWidthProp **=** 600**;**

WindowMinWidthProp **=** 600**;**

WindowHeightProp **=** 500**;**

WindowMinHeightProp **=** 500**;**

Tracks**.**Clear**();**

Users**.**Clear**();**

**}**

**}**

**private** bool UsernameAccessibilityCheck**(**string UserName**)**

**{**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"SELECT \* FROM [accounts] WHERE ([username]='{UserName}')"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

SQLiteDataReader UserExist **=** Command**.**ExecuteReader**();**

**if** **(**UserExist**.**HasRows**)**

**{**

Connect**.**Close**();**

**return** **false;**

**}**

Connect**.**Close**();**

**}**

**return** **true;**

**}**

**private** void CreateAccountFunc**()**

**{**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"INSERT INTO accounts (name, surname, username, password) VALUES (@name, @surname, @username, @password)"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

NameProp **=** char**.**ToUpper**(**NameProp**[**0**])** **+** NameProp**.**Substring**(**1**).**ToLower**();**

Command**.**Parameters**.**AddWithValue**(**"@name"**,** NameProp**);**

SurnameProp **=** char**.**ToUpper**(**SurnameProp**[**0**])** **+** SurnameProp**.**Substring**(**1**).**ToLower**();**

Command**.**Parameters**.**AddWithValue**(**"@surname"**,** SurnameProp**);**

Command**.**Parameters**.**AddWithValue**(**"@username"**,** UsernameProp**);**

Command**.**Parameters**.**AddWithValue**(**"@password"**,** PasswordProp**);**

Connect**.**Open**();**

Command**.**ExecuteNonQuery**();**

Connect**.**Close**();**

**}**

MainLoginWindowState **=** **false;**

CreateAccountWindowState **=** **false;**

AccountCreatedWindowState **=** **true;**

NameProp **=** ""**;**

SurnameProp **=** ""**;**

UsernameProp **=** ""**;**

PasswordProp **=** ""**;**

PasswordConfirmationProp **=** ""**;**

**}**

**private** bool NameSurnameCheck**()**

**{**

NameSurnameWarningAlignmentProp **=** "Left"**;**

**if** **(**string**.**IsNullOrEmpty**(**NameProp**)** **&&** string**.**IsNullOrEmpty**(**SurnameProp**))**

**{**

NameSurnameWarningProp **=** "⚠ Введите имя и фамилию"**;**

**return** **false;**

**}**

**if** **(**string**.**IsNullOrEmpty**(**NameProp**))**

**{**

NameSurnameWarningProp **=** "⚠ Укажите имя"**;**

**return** **false;**

**}**

**if** **(**string**.**IsNullOrEmpty**(**SurnameProp**))**

**{**

NameSurnameWarningProp **=** "⚠ Укажите фамилию"**;**

**return** **false;**

**}**

NameSurnameWarningProp **=** ""**;**

**return** **true;**

**}**

**private** bool UsernameCheck**()**

**{**

UsernameWarningAlignmentProp **=** "Left"**;**

**if** **(**string**.**IsNullOrEmpty**(**UsernameProp**))**

**{**

UsernameWarningProp **=** "⚠ Укажите имя пользователя"**;**

**return** **false;**

**}**

**if** **(**UsernameProp**.**Length **<** 6 **||** UsernameProp**.**Length **>** 30**)**

**{**

UsernameWarningProp **=** "⚠ Имя пользователя должно содержать от 6 до 30 символов."**;**

**return** **false;**

**}**

**if** **(**UsernameProp**[**0**]** **==** '.' **||** UsernameProp**[**UsernameProp**.**Length **-** 1**]** **==** '.'**)**

**{**

**if** **(**UsernameProp**[**0**]** **==** '.'**)**

**{**

UsernameWarningProp **=** "⚠ Первый символ в имени пользователя должен быть латинской буквой (a-z) или цифрой (0-9)."**;**

**}**

**if** **(**UsernameProp**[**UsernameProp**.**Length **-** 1**]** **==** '.'**)**

**{**

UsernameWarningProp **=** "⚠ Последний символ в имени пользователя должен быть латинской буквой (a-z) или цифрой (0-9)."**;**

**}**

**return** **false;**

**}**

**if** **(!**UsernameAccessibilityCheck**(**UsernameProp**))**

**{**

UsernameWarningProp **=** "⚠ Это имя пользователя уже занято. Попробуйте другое."**;**

**return** **false;**

**}**

UsernameWarningProp **=** ""**;**

**return** **true;**

**}**

**private** bool PasswordCheck**()**

**{**

PasswordWarningAlignmentProp **=** "Left"**;**

**if** **(**string**.**IsNullOrEmpty**(**PasswordProp**))**

**{**

PasswordWarningProp **=** "⚠ Введите пароль"**;**

**return** **false;**

**}**

**if** **(**PasswordProp**.**Length **<** 8**)**

**{**

PasswordWarningProp **=** "⚠ Пароль не может быть короче 8 символов"**;**

**return** **false;**

**}**

**if** **(**PasswordProp **!=** PasswordConfirmationProp**)**

**{**

PasswordWarningProp **=** "⚠ Пароли не совпадают. Повторите попытку."**;**

**return** **false;**

**}**

Regex Validate **=** **new** Regex**(**"^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9])(?=.\*?[#?!@$%^&\*-]).{8,}$"**);**

**if** **(!**Validate**.**IsMatch**(**PasswordProp**))**

**{**

PasswordWarningProp **=** "⚠ Пароль недостаточно надежен. Попробуйте сочетание букв, цифр и символов."**;**

**return** **false;**

**}**

PasswordWarningProp **=** ""**;**

**return** **true;**

**}**

**private** void NextCreateAccount**()**

**{**

bool check1 **=** NameSurnameCheck**();**

bool check2 **=** UsernameCheck**();**

bool check3 **=** PasswordCheck**();**

**if** **(**check1 **&&** check2 **&&** check3**)**

**{**

CreateAccountFunc**();**

**}**

**}**

**private** void NextAccountCreated**()**

**{**

MainLoginWindowState **=** **true;**

CreateAccountWindowState **=** **false;**

AccountCreatedWindowState **=** **false;**

**}**

**private** bool UserExistsCheck**()**

**{**

**if** **(**string**.**IsNullOrEmpty**(**UsernameProp**))**

**{**

UserExistsWarningProp **=** "⚠ Введите имя пользователя"**;**

**return** **false;**

**}**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"SELECT \* FROM [accounts] WHERE ([username]='{UsernameProp}')"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

SQLiteDataReader UserExist **=** Command**.**ExecuteReader**();**

**if** **(**UserExist**.**HasRows**)**

**{**

UserExistsWarningProp **=** ""**;**

PasswordEntryWarningProp **=** ""**;**

Connect**.**Close**();**

MainLoginWindowState **=** **false;**

CreateAccountWindowState **=** **false;**

AccountCreatedWindowState **=** **false;**

PasswordEntryWindowState **=** **true;**

**return** **true;**

**}**

**}**

UserExistsWarningProp **=** "⚠ Не удалось найти аккаунт"**;**

**return** **false;**

**}**

**private** bool PasswordEntryCheck**()**

**{**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"SELECT \* FROM [accounts] WHERE (username='{UsernameProp}' AND password='{PasswordEntryProp}')"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

SQLiteDataReader Exist **=** Command**.**ExecuteReader**();**

**if** **(**Exist**.**HasRows**)**

**{**

PasswordEntryWarningProp **=** ""**;**

MainLoginWindowState **=** **false;**

CreateAccountWindowState **=** **false;**

AccountCreatedWindowState **=** **false;**

PasswordEntryWindowState **=** **false;**

**if** **(**UsernameProp **==** "administrator"**)**

**{**

AdminWindowState **=** **true;**

MainWindowState **=** **false;**

WindowWidthProp **=** 600**;**

WindowMinWidthProp **=** 600**;**

WindowHeightProp **=** 500**;**

WindowMinHeightProp **=** 500**;**

Connect**.**Close**();**

DatabaseUsersLoad**();**

**return** **true;**

**}**

MainWindowState **=** **true;**

WindowWidthProp **=** 1000**;**

WindowMinWidthProp **=** 1000**;**

WindowHeightProp **=** 700**;**

WindowMinHeightProp **=** 700**;**

Connect**.**Close**();**

DatabaseLoad**();**

**return** **true;**

**}**

Connect**.**Close**();**

**}**

PasswordEntryWarningProp **=** "⚠ Неверный пароль."**;**

**return** **false;**

**}**

**private** void DatabaseLoad**()**

**{**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"SELECT \* FROM [data] WHERE (owner='{UsernameProp}')"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

SQLiteDataReader reader **=** Command**.**ExecuteReader**();**

Tracks**.**Clear**();**

**while** **(**reader**.**Read**())**

**{**

Tracks**.**Add**(**

**new** Track**()**

**{**

Title **=** reader**[**"title"**].**ToString**(),**

Artist **=** reader**[**"artist"**].**ToString**(),**

Darling **=** Convert**.**ToBoolean**(**reader**[**"darling"**]),**

Cover **=** **new** Avalonia**.**Media**.**Imaging**.**Bitmap**(new** MemoryStream**((**byte**[])**reader**[**"cover"**])),**

ReleaseDate **=** reader**[**"release\_date"**].**ToString**(),**

LastChangeTime **=** reader**[**"last\_change\_time"**].**ToString**()**

**}**

**);**

**}**

Connect**.**Close**();**

**}**

**}**

**private** void DatabaseUpdate**()**

**{**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"DELETE FROM [data] WHERE (owner='{UsernameProp}')"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

**try**

**{**

Command**.**ExecuteNonQuery**();**

**}**

**finally**

**{**

**}**

**for** **(**int i **=** 0**;** i **<** Tracks**.**Count**;** i**++)**

**{**

byte**[]** byteArray **=** **new** byte**[**0**];**

MemoryStream stream **=** **new** MemoryStream**();**

Tracks**[**i**].**Cover**.**Save**(**stream**);**

stream**.**Close**();**

byteArray **=** stream**.**ToArray**();**

commandText **=** $"INSERT INTO [data] (owner, title, artist, darling, cover, release\_date, last\_change\_time) " **+**

$"VALUES ('{UsernameProp}', '{Tracks[i].Title}', '{Tracks[i].Artist}', '{Tracks[i].Darling}', " **+**

$"@cover\_param, '{Tracks[i].ReleaseDate}', '{Tracks[i].LastChangeTime}')"**;**

Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Command**.**Parameters**.**AddWithValue**(**"cover\_param"**,** byteArray**);**

**try**

**{**

Command**.**ExecuteNonQuery**();**

**}**

**finally**

**{**

**}**

**}**

**}**

**}**

**private** void DatabaseUsersLoad**()**

**{**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"SELECT \* FROM [accounts]"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

SQLiteDataReader reader **=** Command**.**ExecuteReader**();**

**while** **(**reader**.**Read**())**

**{**

**if** **(**reader**[**"username"**].**ToString**()** **!=** "administrator"**)**

**{**

Users**.**Add**(**

**new** User**()**

**{**

Username **=** reader**[**"username"**].**ToString**(),**

Selected **=** **false**

**}**

**);**

**}**

**}**

Connect**.**Close**();**

**}**

**}**

**private** void AllTracksLoad**()**

**{**

**using** **(**SQLiteConnection Connect **=** **new** SQLiteConnection**(**@"Data Source=db.sqlite3; Version=3;"**))**

**{**

string commandText **=** $"SELECT \* FROM [data]"**;**

SQLiteCommand Command **=** **new** SQLiteCommand**(**commandText**,** Connect**);**

Connect**.**Open**();**

SQLiteDataReader reader **=** Command**.**ExecuteReader**();**

TracksForReport**.**Clear**();**

**while** **(**reader**.**Read**())**

**{**

**if** **(**TracksForReport**.**Any**(**x**=>**

string**.**Equals**(**x**.**Title**,** reader**[**"title"**].**ToString**(),** StringComparison**.**OrdinalIgnoreCase**)** **&&**

string**.**Equals**(**x**.**Artist**,** reader**[**"artist"**].**ToString**(),** StringComparison**.**OrdinalIgnoreCase**)**

**))**

**{**

int myIndex **=** TracksForReport**.**IndexOf**(**TracksForReport**.**Where**(**x**=>**

string**.**Equals**(**x**.**Title**,** reader**[**"title"**].**ToString**(),** StringComparison**.**OrdinalIgnoreCase**)** **&&**

string**.**Equals**(**x**.**Artist**,** reader**[**"artist"**].**ToString**(),** StringComparison**.**OrdinalIgnoreCase**)).**FirstOrDefault**());**

TracksForReport**[**myIndex**].**Count**++;**

**}**

**else**

**{**

TracksForReport**.**Add**(**

**new** TrackForReport**()**

**{**

Title **=** reader**[**"title"**].**ToString**(),**

Artist **=** reader**[**"artist"**].**ToString**(),**

ReleaseDate **=** reader**[**"release\_date"**].**ToString**(),**

Count **=** 1

**}**

**);**

**}**

**}**

Connect**.**Close**();**

**}**

**}**

**private** void CreateCustomReport**()**

**{**

bool LeastOneSelected **=** **false;**

**for** **(**int i **=** 0**;** i **<** Users**.**Count**;** i**++)**

**{**

**if** **(**Users**[**i**].**Selected**)**

**{**

LeastOneSelected **=** **true;**

**break;**

**}**

**}**

**if** **(!**LeastOneSelected**)**

**{**

**return;**

**}**

ExcelPackage**.**LicenseContext **=** LicenseContext**.**NonCommercial**;**

ExcelPackage excel **=** **new** ExcelPackage**();**

**for** **(**int i **=** 0**;** i **<** Users**.**Count**;** i**++)**

**{**

**if** **(!**Users**[**i**].**Selected**)**

**{**

**continue;**

**}**

var workSheet **=** excel**.**Workbook**.**Worksheets**.**Add**(**Users**[**i**].**Username**);**

UsernameProp **=** Users**[**i**].**Username**;**

DatabaseLoad**();**

workSheet**.**Row**(**1**).**Style**.**HorizontalAlignment **=** ExcelHorizontalAlignment**.**Center**;**

workSheet**.**Row**(**1**).**Style**.**Font**.**Bold **=** **true;**

workSheet**.**Column**(**1**).**Width **=** 20**;**

workSheet**.**Column**(**2**).**Width **=** 20**;**

workSheet**.**Column**(**3**).**Width **=** 20**;**

workSheet**.**Column**(**4**).**Width **=** 20**;**

workSheet**.**Column**(**5**).**Width **=** 20**;**

workSheet**.**Cells**[**1**,** 1**].**Value **=** "Нравится"**;**

workSheet**.**Cells**[**1**,** 2**].**Value **=** "Название"**;**

workSheet**.**Cells**[**1**,** 3**].**Value **=** "Артист"**;**

workSheet**.**Cells**[**1**,** 4**].**Value **=** "Дата релиза"**;**

workSheet**.**Cells**[**1**,** 5**].**Value **=** "Всего треков:"**;**

workSheet**.**Cells**[**1**,** 6**].**Value **=** Tracks**.**Count**;**

**for** **(**int j **=** 0**;** j **<** Tracks**.**Count**;** j**++)**

**{**

workSheet**.**Cells**[**2 **+** j**,** 1**].**Value **=** Tracks**[**j**].**Darling **?** "Да" **:** "Нет"**;**

workSheet**.**Cells**[**2 **+** j**,** 2**].**Value **=** Tracks**[**j**].**Title**;**

workSheet**.**Cells**[**2 **+** j**,** 3**].**Value **=** Tracks**[**j**].**Artist**;**

workSheet**.**Cells**[**2 **+** j**,** 4**].**Value **=** Tracks**[**j**].**ReleaseDate**;**

**}**

**}**

string strPath **=** "Пользовательский отчет.xlsx"**;**

**if** **(**File**.**Exists**(**strPath**))**

**{**

File**.**Delete**(**strPath**);**

**}**

FileStream objFileStrm **=** File**.**Create**(**strPath**);**

objFileStrm**.**Close**();**

File**.**WriteAllBytes**(**strPath**,** excel**.**GetAsByteArray**());**

excel**.**Dispose**();**

**}**

**private** void CreateIntegratedReport**()**

**{**

ExcelPackage**.**LicenseContext **=** LicenseContext**.**NonCommercial**;**

ExcelPackage excel **=** **new** ExcelPackage**();**

var workSheet **=** excel**.**Workbook**.**Worksheets**.**Add**(**"Статистика"**);**

AllTracksLoad**();**

workSheet**.**Row**(**1**).**Style**.**HorizontalAlignment **=** ExcelHorizontalAlignment**.**Center**;**

workSheet**.**Row**(**1**).**Style**.**Font**.**Bold **=** **true;**

workSheet**.**Column**(**1**).**Width **=** 20**;**

workSheet**.**Column**(**2**).**Width **=** 20**;**

workSheet**.**Column**(**3**).**Width **=** 20**;**

workSheet**.**Column**(**4**).**Width **=** 25**;**

workSheet**.**Cells**[**1**,** 1**].**Value **=** "Название"**;**

workSheet**.**Cells**[**1**,** 2**].**Value **=** "Артист"**;**

workSheet**.**Cells**[**1**,** 3**].**Value **=** "Дата релиза"**;**

workSheet**.**Cells**[**1**,** 4**].**Value **=** "У скольки пользователей"**;**

ObservableCollection**<**TrackForReport**>** SortedTracksForReport **=** **new** ObservableCollection**<**TrackForReport**>(**TracksForReport**.**OrderByDescending**(**x **=>** x**.**Count**));**

**for** **(**int i **=** 0**;** i **<** SortedTracksForReport**.**Count**;** i**++)**

**{**

workSheet**.**Cells**[**2 **+** i**,** 1**].**Value **=** SortedTracksForReport**[**i**].**Title**;**

workSheet**.**Cells**[**2 **+** i**,** 2**].**Value **=** SortedTracksForReport**[**i**].**Artist**;**

workSheet**.**Cells**[**2 **+** i**,** 3**].**Value **=** SortedTracksForReport**[**i**].**ReleaseDate**;**

workSheet**.**Cells**[**2 **+** i**,** 4**].**Value **=** SortedTracksForReport**[**i**].**Count**;**

**}**

string strPath **=** "Интегральный отчет.xlsx"**;**

**if** **(**File**.**Exists**(**strPath**))**

**{**

File**.**Delete**(**strPath**);**

**}**

FileStream objFileStrm **=** File**.**Create**(**strPath**);**

objFileStrm**.**Close**();**

File**.**WriteAllBytes**(**strPath**,** excel**.**GetAsByteArray**());**

excel**.**Dispose**();**

**}**

**}**

**}**

Файл «MainWindow.axaml.cs»

**using** Avalonia**.**Controls**;**

**using** Avalonia**.**GUI**.**ViewModels**;**

**using** Avalonia**.**Interactivity**;**

**namespace** Avalonia**.**GUI**.**Views

**{**

**public** **partial** class MainWindow **:** Window

**{**

**public** MainWindow**()**

**{**

InitializeComponent**();**

**}**

**private** **async** void Button\_OnClick**(object?** sender**,** RoutedEventArgs e**)**

**{**

var dialog **=** **new** OpenFileDialog**();**

var result **=** **await** dialog**.**ShowAsync**(this);**

**if** **(**result **!=** **null)**

**{**

**((**MainWindowViewModel**)**DataContext**).**SetImagePath**(**result**[**0**]);**

**}**

**}**

**}**

**}**

**Скриншоты работы программы**

1. Установка программы

Ниже приведены скриншоты работы установочного файла.

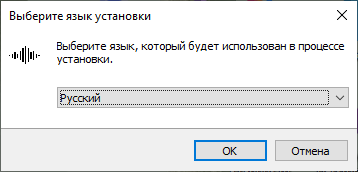


Рисунок 1 – Выбор языка установки.

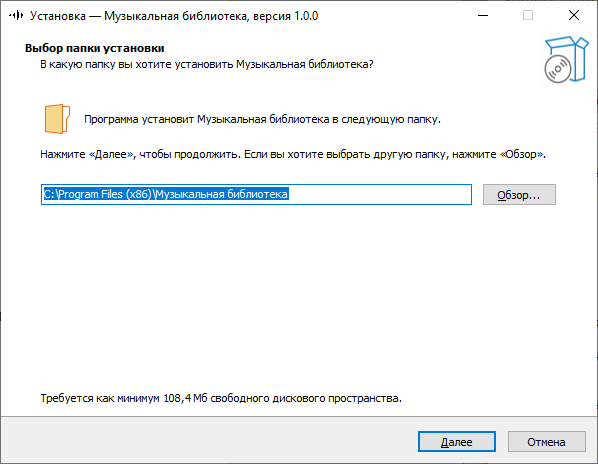


Рисунок 2 – Выбор папки установки.

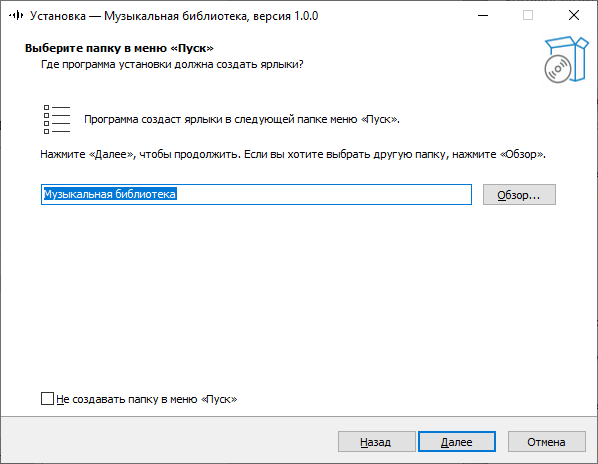


Рисунок 3 – Выбор папки в меню «Пуск».

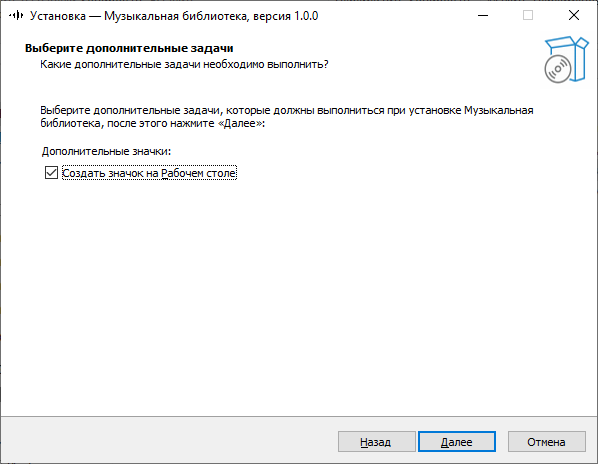


Рисунок 4 – Выбор дополнительных задач.

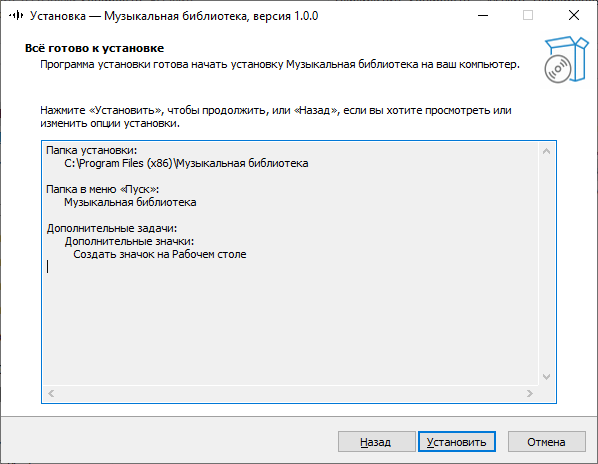


Рисунок 5 – Все готово к установке.

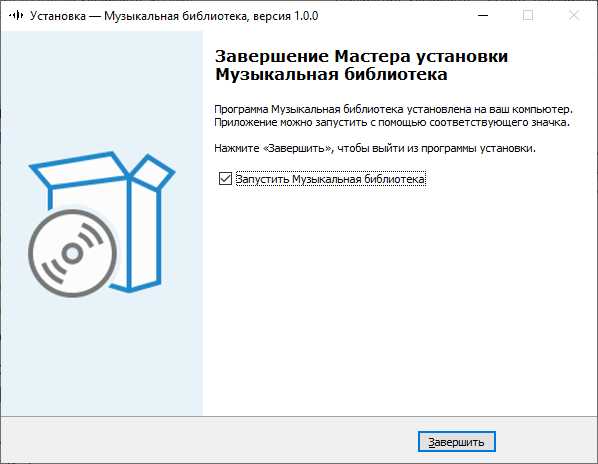


Рисунок 6 – Завершение установки.

**Выводы**

Получены навыки создания установочного файла для программы, написанной на языке C#.

**Список литературы**

1. Avalonia UI Limited. Avalonia Docs [Электронный ресурс]: Learn how to create .NET GUI applications for Windows, macOS, Linux, iOS, Android and Web Assembly from a single codebase. – Электрон. Текстовые дан. – Режим доступа: <https://docs.avaloniaui.net/>, свободный.

2. Васильев А.Н. Программирование на C# для начинающих [Текст] / А.Н. Васильев. – Москва: Бомбора, 2018. – 592 с.